CAR: CONTEXT AWARE ADAPTIVE ROUTING FOR INTERMITTENTLY CONNECTED MOBILE NETWORKS

By

RINI SIMON

Reg. No: 0820108015

of

KUMARAGURU COLLEGE OF TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University, Coimbatore)

COIMBATORE – 641 006

A PROJECT REPORT

Submitted to the

FACULTY OF INFORMATION AND COMMUNICATION ENGINEERING

In partial fulfillment of the requirements
for the award of the degree
of

MASTER OF ENGINEERING
IN

COMPUTER SCIENCE AND ENGINEERING

MAY 2010

BONAFIDE CERTIFICATE

Certified that this project report titled "**CAR: CONTEXT AWARE ADAPTIVE ROUTING FOR INTERMITTENDLY CONNECTED MOBILE NETWORKS**" is the bonafide work of **RINI SIMON (0820108015)** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report of dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

GUIDE

(Mrs.N.RAJATHI)

HEAD OF THE DEPARTMENT

(Dr.S.THANGASAMY)

The candidate with **University Register No. 0820108015** was examined by us in Project Viva-Voce examination held on 17/5/10

Internal Examiner

External Examiner

VIVEKANANDHA
INSTITUTE OF ENGINEERING AND TECHNOLOGY FOR WOMEN
ELAYAMPALAYAM, TIRUCHENGODE.

Department of Computer Applications

NATIONAL CONFERENCE

ON

" EMERGING TECHNOLOGIES IN ADVANCED COMPUTING AND COMMUNICATION "

13 - March, 2010

CERTIFICATE

This is to Certify that Mr. / Ms. / Dr. RINI SIMON of II ME(CSE), Kumaraguru College of Technology has participated in the " National Conference on ETACC '10 ", organized by " VIVACIOUS " the professional association of Computer Applications on 13th March 2010 and presented a paper on CAR: Context Aware Adaptive Routing for Intermitted Connected Mobile Networks.

A. Ahamed Wahab
HOD & Organizing Secretary

V.
Principal

Chairman & Secretary

ABSTRACT

Previous researches in mobile ad hoc network assumed that a connected path exist between sender and receiver at any point of time, but in decentralized system no such path exist between sender and receiver. So delay tolerant network obtained a great attention from research community to fill the gap between ad hoc network research and real applications.

Context Aware Adaptive routing protocol provides a unicast communication in intermittently connected mobile ad hoc networks. Here nodes are used as the carrier of the message among network partitions to achieve successful delivery to the destination. So the main aim is to select a best carrier for transferring the message to the destination. The best carrier is selected by using prediction technique and Utility theory. The decision is based upon the mobility of the host, i.e. a highly mobile host is a good carrier of the message as it meets many hosts, past co-location with the recipient which assumes that the host will meet the recipient again in the near future and finally the battery power of the node. In this proposed system data loss will never occur and delivery of data is guaranteed.

ஆய்வுச்சுருக்கம்

இன்றைய காலக்கட்டதில் ஒரு பிணையத்தில் தகவல் பரிமாறுதல் மிக கவனமாகவும் ரகசியமாகவும் இருக்க வேண்டும் இல்லையென்றால் அந்த தகவல்கள் மற்றவர்களுக்கு வெளிப்பட்டு தவரான மாறுதல்கள் எற்பட வாய்ப்புண்டு.

இந்த ஆய்வில் இரண்டு கம்பி இல்லா பிணையங்களுக்கு இடையில் தரவுகளை பரிமாற்றும் செயலில் ஈடுபடுகிறது. ஒரு கம்பி இல்லா பிணையத்தில் அனுப்புநர் தரவுகளை குறைந்த தொலைதூரத்தில் இருக்கும் சிறப்பு தரவை உள்ளடக்கும் முடிச்சுக்கு அனுப்பும் பின்பு அந்த முடிச்சு தரவை எடுத்து கொண்டு அடுத்து இருக்கும் பிணையத்திற்க்கு எடுத்து செல்லும்.

சிறப்பு தரவை உள்ளடக்கும் முடிச்சு முதல் பிணையத்தை விட்டு அடுத்த பிணையத்தில் சென்று அந்த பிணையத்தோடு தொடர்பு உண்டாக்கி அனுப்புநர் தந்த தரவை குறைந்த தொலைதூரத்தில் பெறுநர்க்கு சென்று சேர்க்கும்.

ACKNOWLEDGEMENT

I express my profound gratitude to our Chairman **Padmabhusan Arutselver Dr. N. Mahalingam B.Sc, F.I.E** for giving this great opportunity to pursue this course.

I would like to begin by thanking to **Dr.S.Ramachandran**, *principal* for providing the necessary facilities to complete my thesis.

I take this opportunity to thank **Dr. S.Thangasamy,** Dean ,*Head of the Department*, Computer Science and Engineering, for his precious suggestions.

I register my hearty appreciation to **Mrs.N.Rajathi M.E,** *Assistant professor,* my thesis advisor. I thank for her support, encouragement and ideas. I thank her for the countless hours she has spent with me, discussing everything from research to academic choices.

I thank all project committee members for their comments and advice during the reviews. Special thanks to **Mrs.V.Vanitha M.E (Ph.D),** *Assistant professor*, Department of Computer science and Engineering, for arranging the brain storming project review sessions.

I would like to convey my honest thanks to all **Teaching** staff members and **Non Teaching** staffs of the department for their support. I would like to thank all my classmates who gave me a proper light moments and study breaks apart from extending some technical support whenever I needed them most.

I dedicate this project work to my **parents** for no reasons but feeling from bottom of my heart, without their love this work wouldn't possible.

TABLE OF CONTENTS

CONTENTS **PAGE NO**

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CAR	Context Aware Adaptive Routing For Intermittently Connected Mobile Networks
MANET	Mobile Ad Hoc Network
DTN	Delay Tolerant Network
DSDV	Destination Sequence Distance Vector Routing
ODMP	On-Demand Multicast Routing Protocol
MF	Message Ferrying
NIMF	Node Initiated Message Ferrying
FIMF	Ferry Initiated Message Ferrying
PROPHET	Probabilistic Routing Technique
CP	Concentration Points

LIST OF SYMBOLS

U	Utility Function
w	Weight of parameter
$\hat{U}$	Utility at time t+T

CHAPTER 1

INTRODUCTION

In the recent years wireless networks have witnessed a tremendous increase of popularity in both research and industry. There are currently two variations of mobile networks. The first is widely known as infrastructure networks since the gateways that connect them to other networks (like the internet) are fixed and wired. The bridges in these networks are also known as base stations. In an environment like this, a node is able to roam freely and establish a connection link with the nearest base station that is within its communication range. As the mobile node moves out of the base station that it was connected with, it falls into the range of another and hand off occurs between the old base station and the current one, enabling the mobile unit to continue communication seamlessly through the network. These types of networks are most widely applied in office areas and include the Wireless Local Area Networks (WLANS).

The second type of wireless networks is the infrastructure less mobile network that is also known as an ad hoc network. Infrastructure less mobile network has no fixed routers and base stations and the participating nodes are capable of movement. Due to the limited transmission range, multiple hops may be required for nodes to communicate across the ad hoc network. Routing functionality is incorporated into each host, thus ad hoc networks can be characterized as having dynamic, multi-hop and constantly changing topologies. These types of network are excellent for sharing information in areas lacking of communication infrastructures such as disaster areas, thick forest or unexplored territories (e.g. deep space or deep sea).

1.1. MOBILE AD HOC NETWORKS

There are many situations where users of a network cannot rely on infrastructure, it is too expensive or there is none at all. In these situations mobile ad hoc network is the only choice. Mobile ad hoc networks are a new paradigm of wireless communication for mobile hosts (which we call nodes). An ad hoc network is a collection of two or more devices equipped with wireless communications and networking capability. In an ad hoc network, there is no fixed infrastructure such as base stations or mobile switching centers. A Mobile Ad Hoc network is a self-configurable, self organizing, infrastructure less multi-hop mobile wireless network. Mobile nodes that are within each other's radio range communicate directly via wireless links, while those that are outside their radio range, use an intermediate node to relay or forward the packet from the source towards the destination. Self-configuring and self-organizing means, that a network can be formed on the fly without the need for any system administration.

A Mobile Ad Hoc Network (MANET), sometimes called a mobile mesh network, is a self-configuring network of mobile devices connected by wireless links. Each device in a MANET is free to move independently in any direction, and will therefore change its link to other devices frequently. Each must forward traffic unrelated to its own use, and therefore be a router. The primary challenge in building a MANET is equipping each device to continuously maintain the information required to properly route traffic. Such networks may operate by themselves or may be connected to the larger internet. MANETs are a kind of wireless ad hoc networks that usually has a routable networking environment on top of a Link layer ad hoc network. They are also a type of mesh network, but many mesh networks are not mobile or not wireless. A node in the mobile ad-hoc network communicates directly with nodes within wireless range as well as indirectly with all other destinations using a dynamically determined multi-hop route through other nodes in the MANET.

Each node in a MANET is capable of moving independently, thus the network topology can change continuously and dramatically. Each node also functions as a router that discovers and maintains routes to other nodes and forwards packets for other nodes. A MANET can be deployed without any wired base stations or infrastructure support. These features make a MANET very attractive or reliable, but fast network establishment and self-reconfiguration are required. Primary challenge is that each device has to continuously maintain the information required to properly route the traffic. Types of MANET include VANET (Vehicular Ad hoc Network), Intelligent Vehicular Ad hoc Network, internet based Mobile Ad hoc Network (iMANET).

The Growth of laptops and 802.11/Wi-Fi wireless networking has made MANETs a popular research topic since the mid-to late 1990s.Many academic papers evaluate protocols and abilities assuming varying degree of mobility within a bounded space, usually with all nodes within a few hops of each other and usually with nodes sending data at a constant rate. Different protocols are then evaluated based on the packet drop rate, the overhead introduced by the routing protocol, and other measures.

1.2. PROPOSED SYSTEM [1]

The Context-aware Adaptive Routing (CAR) protocol, an approach to delay-tolerant mobile ad hoc network routing that uses prediction to allow the efficient routing of messages to the recipient. A host willing to send a message to a recipient or any host in the multi hop path to choose the best next hop (or carrier) for the message. The decision is based on the mobility of the host (a highly mobile host is a good carrier as it meets many hosts) and its past colocation with the recipient (we implicitly assume that past colocation indicates that the host will meet the recipient again in the future). CAR does not require any previous knowledge of the routes of the hosts like other approaches such as the Message Ferrying project that rely on the a priori knowledge of the routes of the special hosts carrying the information.

The advantages of CAR protocol includes it provides communication in intermittently connected mobile ad hoc network, the key problem solved by the protocol is selection of carrier. CAR is able to deliver messages synchronously (i.e., without storing them in buffers of intermediate nodes when there are no network partitions between the sender and the receiver) and asynchronously (i.e., by means of a store-and-forward mechanism when there are partitions). The delivery process depends on whether or not the recipient is present in the same connected region of the network (cloud) as the sender. If both are currently in the same connected portion of the network, the message is delivered using an underlying synchronous routing protocol to determine a forwarding path. If a message cannot be delivered synchronously, the best carriers for a message are those that have the highest chance of successful delivery, i.e., the highest delivery probabilities.

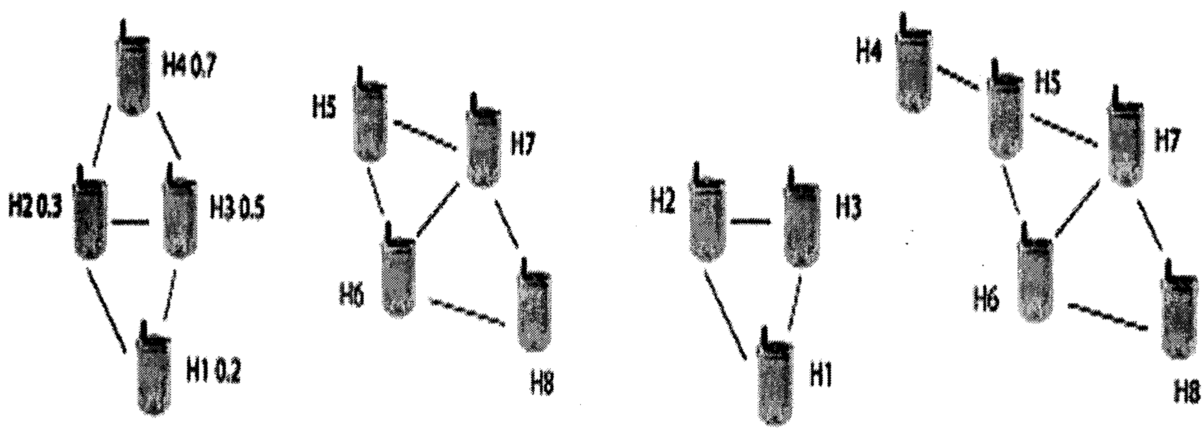


Fig 1.1.a **Fig 1.1.b**

Fig 1.1.Two connected cloud with associated delivery probability for message transmission between H1 and H8

If node h1 wants to send message to h4 a connected path between sender and receiver exist, so the message will be directly send to the destination after finding the shortest path.

The operation of the CAR protocol is as follows, two groups of nodes are connected as in Fig1.1.The Dynamic Destination-Sequenced Distance Vector (DSDV) is used to support synchronous routing. Host H1 wishes to send a message to H8.This cannot be done synchronously, because there is no connected path between the two. Suppose the delivery probabilities for H8 are as shown in Fig1.1.a.In this case, the host possessing the best delivery probability to host H8 is H4. Consequently, the message is sent to H4, which stores it. After a certain period of time, H4 moves to the other cloud as in Fig. 1.1.b. Since a connected path between H4 and H8 now exists, the message is delivered to its intended recipient. Using DSDV, for example, H4 is able to send the message shortly after joining the cloud, since this is when it will receive the routing information relating to H8.

CHAPTER 2

LITERATURE SURVEY

2.1. MANET FEATURES

Some of the salient features that describe the MANET clearly are

- **Autonomous terminal:** In MANET, each mobile terminal is an autonomous node, which may function both as a host and a router (to perform switching functions).
- **Distributed operation:** Since there is no background network, the control and management of the network is distributed among the terminals.
- **Multi-hop routing:** When delivering data packets from a source to its destination (i.e. only when the nodes are not directly linked), the packets should be forwarded via one or more intermediate nodes.
- **Dynamic network topology:** Since the nodes are mobile, the network topology may change rapidly and unpredictably and the connectivity among the terminals may vary with time.
- **Light-weight terminals:** In most cases, the MANET nodes are mobile devices with less CPU processing capability, small memory size, and low power storage. Such devices need optimized algorithms and mechanisms that implement the computing and communication functions.

2.2. MANET APPLICATIONS

With the increase of the portable devices as well as progress in wireless communication, ad hoc networking is gaining importance with the increasing number of widespread applications. Ad hoc networking can be applied anywhere, where there is little or no communication infrastructure or the existing infrastructure is expensive or inconvenient to use.

Typical applications include:

- **Military battlefield**: Ad hoc networking would allow the military to take advantage of a common place network technology to maintain an information network between the soldiers, vehicles, and military information head quarters.
- **Commercial Sector**: Emergency rescue operations (like fire, flood, earthquake, etc.) must take place where non-existing or damaged communications infrastructure and rapid deployment of a communication network is needed.
- **Local Level**: Ad hoc networks can autonomously link an instant and temporary multimedia network using notebook computers or palmtop computers to spread and share information among participant, e.g. conference or classroom.
- **Personal Network Area (PAN)**: Short-range MANET can simplify the intercommunication between various mobile devices (such as PDA, a laptop, a cellular phone).

2.3. CHALLENGES FACED IN MANET

Regardless of the attractive applications, the features of MANET introduce several challenges that must be studied carefully before a wide commercial deployment can be expected. These include:

- **Routing:** Since the topology of the network is continuously changing, the issue of routing packets between any pair of nodes becomes a challenging task. Most protocols should be based on reactive routing instead of proactive.
- **Security and Reliability:** An Ad Hoc network has its particular security problems due to nasty neighbor relaying packets. Further, wireless link

characteristics introduce reliability problems, because of limited wireless transmission range, the broadcast nature of wireless medium (eg:hidden terminal problem),mobility introduced packet losses and data transmission errors.

- **Quality of service (QoS):** Providing different quality of service levels in a constantly changing environment will be a challenge.
- **Internetworking:** The coexistence of routing protocols, for the sake of internetworking a MANET with a fixed network, in a mobile device is a challenge for the mobility management.

MANET is a highly dynamic environment, so the traditional well established multicasting protocols cannot be deployed directly to it. Some modification and extension should be made while considering all the constraints, such as dynamic network topology, limited bandwidth and power. The new protocols should avoid global flooding, should dynamically build the routes and should update both routes and memberships periodically.

In existing work mobile ad hoc network research has often assumed that a connected path exists between a sender and a receiver node at any point in time. This assumption reveals itself unrealistic in many decentralized mobile network applications. To answer this dichotomy, Delay-Tolerant Networking (DTN) has received considerable attention from the research community in recent years as a means of addressing exactly the issue of routing messages in partitioned networks. DTNs span very challenging application scenarios where nodes (e.g. people and wild animals) move around in environments where infrastructures cannot be installed.

Some solutions to routing have been presented also for these cases, starting from the basic epidemic routing where messages are blindly stored and forwarded to all neighboring nodes, generating a flood of messages. The drawback of epidemic dissemination lies in the very high number of messages that are needed to obtain a successful delivery to the right recipient. Other solutions have been proposed to tackle the problem of routing in (possibly mobile) Delay-Tolerant Networks, based on the previous knowledge of the routes of the potential carriers or on probabilistic approaches. The disadvantages the basic epidemic routing are messages are blindly stored and forwarded to all neighboring nodes, generating a flood of messages. The drawback of epidemic dissemination lies in the very high number of messages that are needed to obtain a successful delivery to the right recipient. The problem of routing in (possibly mobile) Delay-Tolerant Networks, based on the previous knowledge of the routes of the potential carriers or on probabilistic approaches.

2.4. EPIDEMIC ROUTING [2]

This routing mechanism is bon routing. Continuously replicate and transmit message to newly discovered contact that do not posses a copy of the message. Random pair-wise exchanges of messages among mobile hosts ensure eventual message delivery. The goals of Epidemic Routing are to i) Maximize message delivery rate ii) Minimize message latency and iii) Minimize the total resources consumed in message delivery.

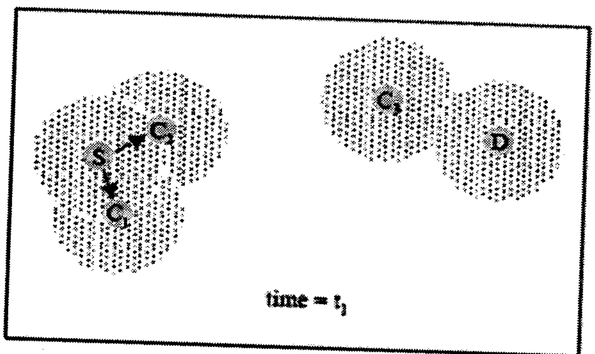


Fig 2.1.a

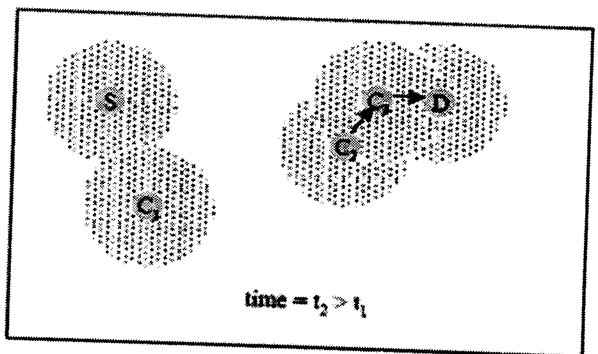


Fig 2.1.b

Fig 2.1.a No connected path between sender and receiver, Fig 2.1.b delivering message to destination

Mobile nodes represented as dark circles and their wireless communication range shown as a dotted circle extending from the source. A source S, wishes to send a message to a destination D, but no connected path is available from S to D. S transmits its messages to its two neighbours C1 and C2, within direct communication range. At some later time, as shown in Figure 2.1.b, C2 comes into direct communication range with another host, C3, and transmits the message to it. C3 is in direct range of D and finally sends the message to its destination

The message exchange in epidemic routing protocol will be as shown in fig 2.2. Host A comes into contact with Host B and initiates an anti-entropy session. During anti-entropy, the two hosts exchange their summary vectors to determine which messages stored remotely have not been seen by the local host. In turn, each host then requests copies of messages that it has not yet seen. The receiving host maintains total autonomy in deciding whether it will accept a message. In step one, A transmits it summary vector SVA to B. SVA is a compact representation of all the messages being buffered at A. In Next step B performs a logical AND operation between the negation of its summary vector SVB (the negation of B's summary vector, representing the messages that it needs) and SVA. That is B determines the set difference between the messages buffered at A and the messages buffered locally at B.

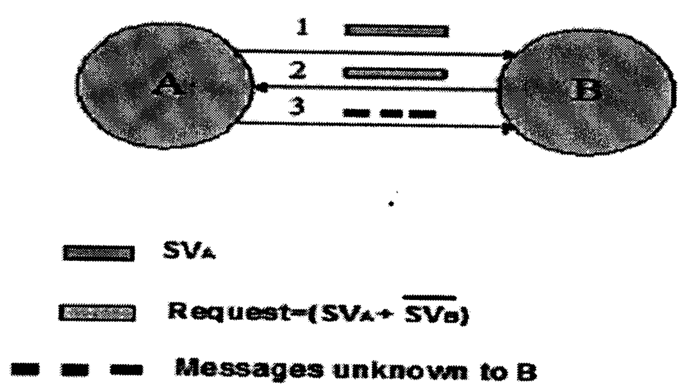


Fig 2.2 Epidemic routing protocol-hosts A and B comes into the transmission range of one another.

It then transmits a vector requesting these messages from A. In step three A transmits the requested messages to B. This process is repeated transitively when B comes into contact with a new neighbour. Given sufficient buffer space and time, these anti-entropy sessions guarantee eventual message delivery through such pair-wise message exchange.

2.5. PROPHET [11]

Probabilistic Routing Technique exploits non randomness of nodes. A probabilistic metric called delivery predictability, *P(a,b) € [0,1]*, at every node *a* for each known destination *b*. This indicates how likely that this node will be able to deliver a message to that destination. When two nodes meet, they exchange summary vectors which contain the delivery predictability information stored at the nodes. This information is used to update the internal delivery predictability vector and then the information in the summary vector is used to decide which messages to request from the other node based on the forwarding strategy used.

2.5.1. Delivery Predictability Calculation

To update the probabilistic metric whenever a node is encountered,

$$P(a,b)=P(a,b)_{old}+(1-P(a,b)_{old})*P_{init}$$

where,

Pinit € [0, 1] is an initialization constant

a -source node

b -destination node

If a pair of nodes does not encounter each other in a while, they are less likely to be good forwarders of messages to each other, thus the delivery predictability values must age, being reduced in the process.

The aging equation,

$$P(a,b)=P(a,b)_{old}*\gamma^k$$

where,

γ is the aging constant

k is the number of time units that have elapsed since the last time the metric was aged.

The delivery predictability also has a transitive property, that is based on the observation that if node A frequently encounters node B, and node B frequently encounters node C, then node C probably is a good node to forward messages destined for node A

$$P(a,c)=P(a,c)_{old}+(1-P(a,c)_{old})*P(a,b)*P(b,c)*\beta$$

where,

β is a scaling constant that decides how large impact the transitivity should have on the delivery predictability.

2.6. Message Ferrying [10]

Message Ferrying (MF) is designed for sparse networks. MF is a proactive mobility assisted approach which utilizes a set of special mobile nodes called message ferries (or ferries) to provide communication services for nodes in the network. Message ferries move around the deployment area and take responsibility for carrying data between nodes. The main idea behind the Message Ferrying approach is to introduce non-randomness in the movement of nodes and exploit such non-randomness to deliver the data. Message ferrying can be used effectively in a variety of applications including battlefields, disaster relief, wide area sensing, non-interactive Internet access and anonymous communication.

Two variations of the MF schemes, depending on whether ferries or the nodes initiate non-random proactive movement. In the Node-Initiated MF (NIMF) scheme, ferries move around the deployed area according to known routes and communicate with other nodes they meet. With knowledge of ferry routes, nodes periodically move close to a ferry and communicate with the ferry. In the Ferry-Initiated MF (FIMF) scheme, ferries move proactively to meet nodes. When a node wants to send packets to other nodes or receive packets, it generates a service request and transmits it to a chosen ferry using a long range radio signal. Upon reception of a service request, the ferry will adjust its trajectory to meet up with the node and exchange packets using short range radios. In both schemes, nodes can communicate with distant nodes that are out of range by using ferries as relays.

FIMF

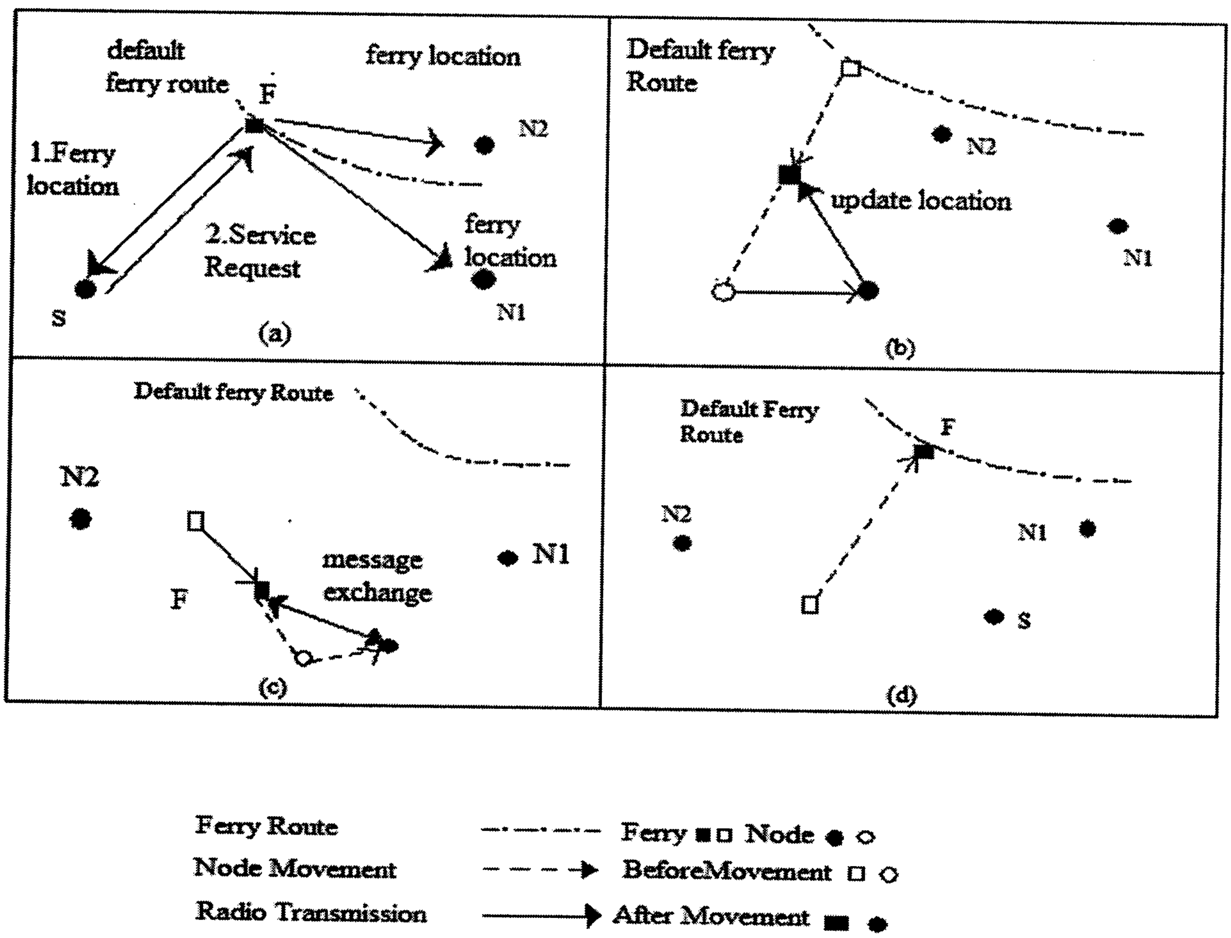


Fig 2.4 simplified example of how the FIMF scheme operates

Initially the ferry *F* follows a specific default route and periodically broadcasts its location to nodes using a long range radio. When a node *S* finds the ferry is nearby and wants to sender receive messages via the ferry, it sends a Service Request message to the ferry using its long range radio as shown in Fig.2.4.a. This message contains the node's location information. Upon reception of a request message, the ferry adjusts its trajectory to meet the node. To guide the ferry movement, the node occasionally transmits Location Update messages to notify the ferry of its new location (Fig. 2.4.b). When the ferry and the node are close enough, they exchange messages via short range radios (Fig. 2.4.c). After completing message exchange with the node, the ferry moves back to its default route as shown in Fig.2.4.d.

NIMF

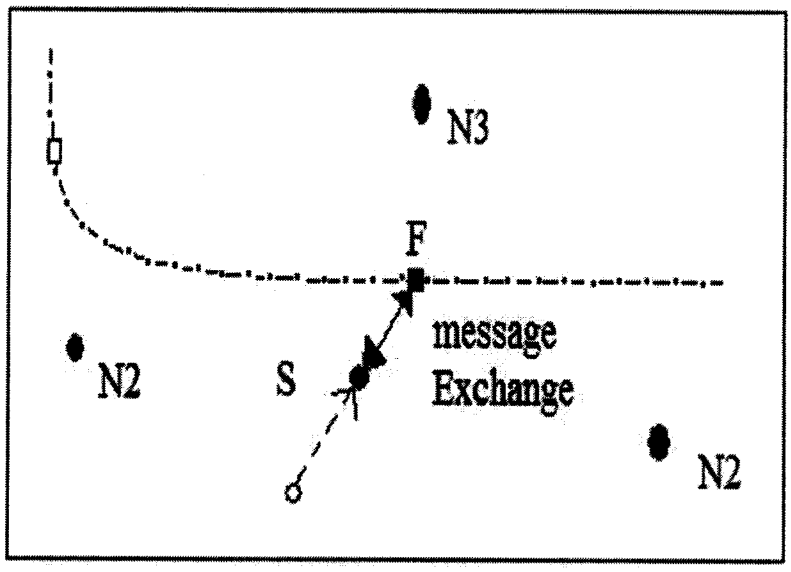


Fig.2.3.a

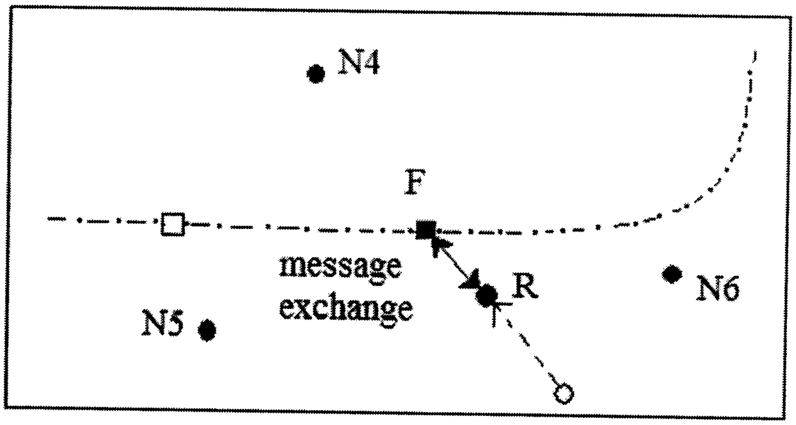


Fig 2.3.b

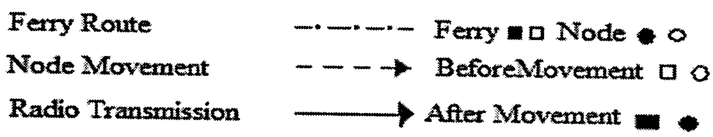


Fig2.3. Simplified example of how the NIMF scheme operates

In Fig.2.3.a, the ferry *F* moves on a known route. As the sending node *S* approaches the ferry, it forwards its messages to the ferry which will be responsible for delivery. In Fig. 2.3.b, the receiving node *R* meets the ferry and receives its messages. By using the ferry as a relay, *S* can send a message to R even there is no end-to-end path between them.

2.7. Island Hopping [3]

In wireless ad hoc network instantaneous end-to-end routes do not always exist, message has to be temporarily buffered at intermediate nodes for the availability of the next link towards the destination. This approach has been referred to as mobility-assisted forwarding, or also as store-carry-forward. Network partitions arise because the distribution of nodes in space is heterogeneous. Network possesses concentration points (CPs), i.e., regions where the node density is much higher than on average, and where nodes have therefore a much better chance of being connected to other nodes than on average. The CPs, and the average flows of nodes between CPs, remains stable over long time-scales. This technique uses mobility assisted forwarding that makes use of stable concentration points.

To achieve this goal, we make three distinct contributions:(i) we introduce a mobility model that explicitly embodies CPs, and justify it through an analysis of a large mobility trace; (ii) we describe the Island Hopping (IH) algorithm that forwards messages through mobility; and (iii) we describe how a collection of mobile nodes can infer the CP topology without any explicit signals from the environment, such as GPS coordinates or beaconing signals. Finally, we summarize these three contributions.

CHAPTER 3
PROBLEM DEFINITION

Mobile ad-hoc network research has often assumed that a connected path exists between sender and receiver node at any point in time. This assumption reveals itself unrealistic in many decentralized mobile network applications such as vehicular networks, wildlife monitoring sensor networks, deep space communication systems and emergency operation networks. To answer this dichotomy, Delay-Tolerant Networking (DTN) has received considerable attention from the research community in recent years as a means of addressing exactly the issue of routing messages in partitioned networks.

DTN span very challenging application scenarios where nodes move around in infrastructures cannot be installed. Some solutions to routing have been presented, starting from the basic epidemic routing where messages are blindly stored and forwarded to all neighbouring nodes generating a flood of messages. The drawback of epidemic dissemination lies in the very high number of messages that are needed to obtain a successful delivery to the right recipient. Other solutions have been proposed to tackle the problem of routing in Delay Tolerant Networks, based on previous knowledge of the routes of the potential carriers or on probabilistic approaches.

Context-Aware Adaptive Routing (CAR) protocol, an approach to delay tolerant mobile ad hoc network routing which uses prediction to allow the efficient routing of messages to the recipient, or any host in the multi hop path to it, uses a Kalman Filter Prediction and multi-criteria decision theory to choose the best next hop for the message. The decision is based on mobility of the host and its past colocation with the recipient.

CHAPTER 4

DESIGN OF THE PROPOSED METHODOLOGY

4.1. SYSTEM ARCHITECTURE

A system architecture or systems architecture is the conceptual design that defines the structure and/or behavior of a system.An architecture description is a formal description of a system, organized in a way that supports reasoning about the structural properties of the system. It defines the system components or building blocks and provides a plan from which products can be procured, and systems developed, that will work together to implement the overall system.

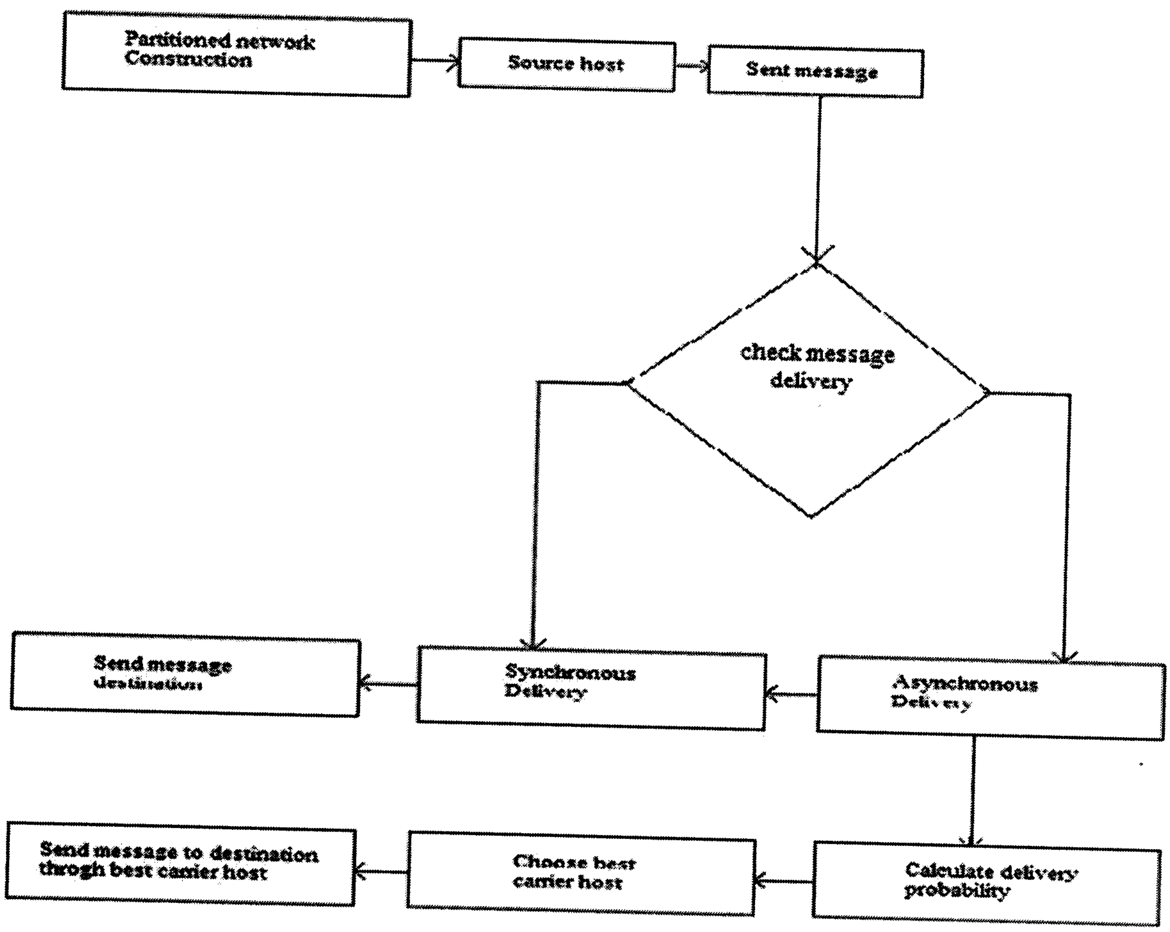


Fig.4.1.System Architecture

4.2.Sequence Diagram

Sequence diagrams model the flow of logic within the system in a visual manner, enabling the document and validate the logic, and are commonly used for both analysis and design purposes. Sequence diagrams are the most popular UML artifact for dynamic modeling, which focuses on identifying the behavior within the system.

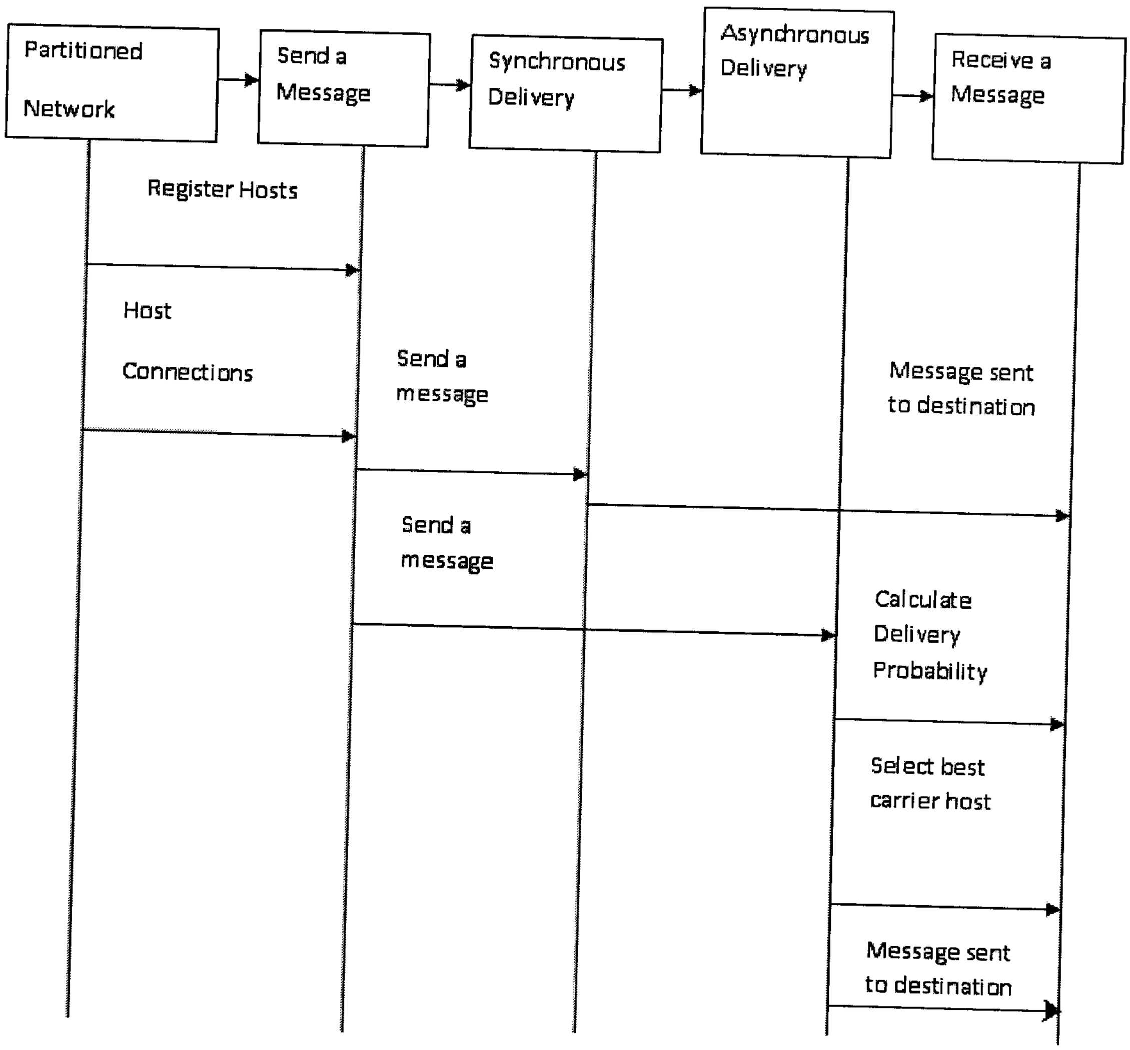


Fig.4.2.Sequence Diagram

4.3.Use Case Diagram

Use case diagrams overview the usage requirements for a system. They are useful for presentations to management or project stakeholders, but for actual development you will find that use cases provide significantly more value because they describe "the meat" of the actual requirements.

Use cases. A use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse

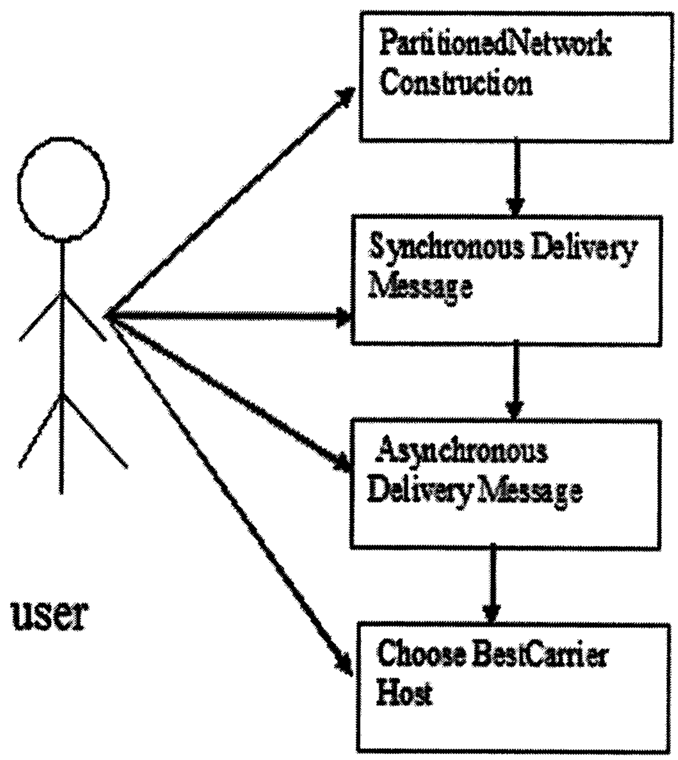


Fig.4.3.Usecase Diagram

4.4.KALMAN FILTER [12]

The Kalman filter is a mathematical method named after Rudolf E. Kalman. Its purpose is to use measurements that are observed over time that contain noise (random variations) and other inaccuracies, and produce values that tend to be closer to the true values of the measurements and their associated calculated values. The Kalman filter has many applications in technology, and is an essential part of the development of space and military technology. Perhaps the most commonly used type of very simple Kalman filter is the phase-locked loop, which is now ubiquitous in FM radios and most electronic communications equipment. Extensions and generalizations to the method have also been developed.

The Kalman filter produces estimates of the true values of measurements and their associated calculated values by predicting a value, estimating the uncertainty of the predicted value, and computing a weighted average of the predicted value and the measured value. The most weight is given to the value with the least uncertainty. The estimates produced by the method tend to be closer to the true values than the original measurements because the weighted average has a better estimated uncertainty than either of the values that went into the weighted average.

4.4.1. Overview of calculation

The Kalman filter uses a system's dynamics model (i.e. physical laws of motion), known control inputs to that system, and measurements (such as from sensors) to form an estimate of the system's varying quantities (its state) that is better than the estimate obtained by using any one measurement alone. As such, it is a common sensor fusion algorithm.

All measurements and calculations based on models are estimates to some degree. Noisy sensor data, approximations in the equations that describe how a system changes, and external factors that are not accounted for introduce some uncertainty about the inferred values for a system's state. The Kalman filter averages a prediction of a system's state with a new measurement using a weighted average. The purpose of the weights is that values with better estimated uncertainty are "trusted" more. The weights are calculated from the covariance, a measure of the estimated uncertainty of the prediction of the system's state. The result of the weighted average is a new state estimate that lies in between the predicted and measured state, and has a better estimated uncertainty than either alone. This process is repeated every time step, with the new estimate and its covariance informing the prediction used in the following iteration. This means that the Kalman filter works recursively and requires only the last "best guess" - not the entire history - of a system's state to calculate a new state.

When performing the actual calculations for the filter (as discussed below), the state estimate and covariance's are coded into matrices to handle the multiple dimensions involved in a single set of calculations. This allows for representation of linear relationships between different state variables (such as position, velocity, and acceleration) in any of the transition models or covariances.

4.5. DESTINATION SEQUENCE DISTANCE VECTOR ROUTING [4]

Destination-Sequenced Distance-Vector Routing (DSDV) is a table-driven routing scheme for ad hoc mobile networks. It was developed by C. Perkins and P.Bhagwat in 1994. The main contribution of the algorithm was to solve the Routing Loop problem. Each entry in the routing table contains a sequence number, the sequence numbers are generally even if a link is present; else, an odd number is used. The number is generated by the destination, and the emitter needs to send out the next update with this number. Routing information is distributed between nodes by sending full dumps infrequently and smaller incremental updates more frequently.

A routing loop is a common problem with various types of networks, particularly computer networks. They are formed when an error occurs in the operation of the routing algorithm, and as a result, in a group of nodes, the path to a particular destination forms a loop.

In the simplest version, a routing loop of size two as shown in fig 4.4, node A thinks that the path to some destination (call it C) is through its neighboring node, node B. At the same time, node B thinks that the path to C starts at node A.

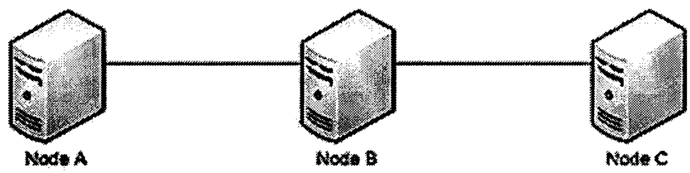


Fig .4.4.Three nodes connected to show DSDV protocol

For example the routing table of Node A in this network is

Destination	Next Hop	Number of Hops	Sequence Number
A	A	0	A 46
A	B	1	B 36
A	C	2	C 28

Table 4.1.Routing table for DSDV of node A using DSDV

This scheme is quite suitable for creating ad hoc networks with small number of nodes.

Disadvantage includes it require a regular update of its routing tables, which uses up battery power and a small amount of bandwidth even when the network is idle. Whenever the topology of the network changes, a new sequence number is necessary before the network re-converges; thus, DSDV is not suitable for highly dynamic networks. (As in all distance-vector protocols, this does not perturb traffic in regions of the network that are not concerned by the topology change.

4.6. ODMRP PROTOCOL [13]

ODMRP (On-Demand Multicast Routing Protocol), is a mesh-based, rather than a conventional tree based, multicast scheme and uses a forwarding group concept (only a subset of nodes forwards the multicast packets via scoped flooding). It applies on-demand procedures to dynamically build routes and maintain multicast group membership. ODMRP is well suited for ad hoc wireless networks with mobile hosts where bandwidth is limited, topology changes frequently, and power is constrained.

4.6.1. Multicast Route and Membership Maintenance

In ODMRP, group membership and multicast routes are established and updated by the source on demand. Protocol comprises a request phase and a reply Phase. When a multicast source has packets to send, it periodically broadcasts to the entire network a member advertising packet, called a JOIN REQUEST. This periodic transmission refreshes the membership information and updates the route. When a node receives a non-duplicate JOIN REQUEST, it stores the upstream node ID (i.e., backward learning) and rebroadcasts the packet. When the JOIN REQUEST packet reaches a multicast receiver, the receiver creates or updates the source entry in its Member Table. While valid entries exist in the Member Table, JOIN TABLES are broadcasted periodically to the neighbours. When a node receives a JOIN TABLE, it checks if the next node ID of one of the entries matches its own ID. If it does, the node realizes that it is on the path to the source and thus is part of the forwarding group. It then sets the FG Flag and broadcasts its own JOIN TABLE built upon matched entries. The JOIN TABLE is thus propagated by each forwarding group member until it reaches the multicast source via the shortest path. This process constructs (or updates) the routes from sources to receivers and builds a mesh of nodes, the forwarding group.

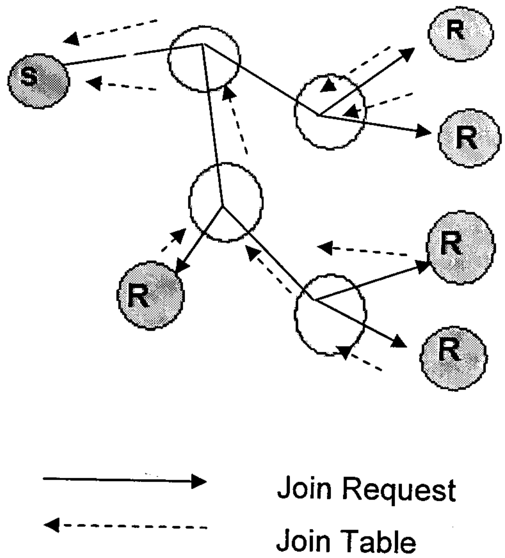


Fig 4. 5. on-Demand Procedure for Membership Setup and Maintenance.

4.6.2. Unicast Capability

One of the major strengths of ODMRP is its unicast routing capability. ODMRP can work with any unicast routing protocol, it can function as both multicast and unicast.

CHAPTER 5

IMPLEMENTATION

5.1. SYSTEM SPECIFICATION

The software requirement specification is produced at the culmination of the analysis task. The function and performance allocated to software as part of system engineering are refined by establishing a complete information description as functional representation, a representation of system behavior, an indication of performance requirements and design constraints, appropriate validation criteria.

Software Requirements

- Language : Java 1.5 or more
- Front End Tool : Swing
- Back End Tool : Sql Server
- Operating System : Windows Xp

Hardware Requirements

- Hard disk : 80 GB
- SDRAM : 256 MB
- Processor : Intel Pentium IV
- Processor Speed : 2 GHz

5.2. SOFTWARE DESCRIPTION

5.2.1. CORE JAVA:

Java can be used to create two types of programs: application and applet. An application is a program that runs on your computer, under the operating system of that computer. That is, an application created by java is more or less like one created using C or C++. An applet is an application designed to be transmitted over the internet and executed by a java-compatible Web Browser. An applet is actually a tiny java program, dynamically downloaded across the network, just like an image, sound file, or video clip. The important difference is that an applet is an intelligent program, not just an animation or media file. In other words, an applet is a program that can react to user input and dynamically change-not just run the same animation or sound over and over.

Java having a major role in internet and the intranet application. The reason for this is Java expends the universe of objects that can move about freely in cyberspace.

5.2.2. Swing

Swing was developed to provide a more sophisticated set of GUI components than the earlier Abstract Window Toolkit. Swing provides a native look and feel that emulates the look and feel of several platforms, and also supports a pluggable look and feel that allows applications to have a look and feel unrelated to the underlying platform.Swing introduced a mechanism that allowed the look and feel of every component in an application to be altered without making substantial changes to the application code. The introduction of support for a pluggable look and feel allows Swing components to emulate the appearance of native components while still retaining the benefits of platform independence.

5.2.3. SQL SERVER

SQL Server is a relational database management system (RDBMS) from Microsoft that's designed for the enterprise environment. SQL Server runs on T-SQL (Transact -SQL), a set of programming extensions from Sybase and Microsoft that add several features to standard SQL, including transaction control, exception and error handling, row processing, and declared variables.

ODBC

Microsoft Open Database Connectivity (ODBC) is a standard programming interface for application developers and database systems providers. Before ODBC became a *de facto* standard for Windows programs to interface with database systems, programmers had to use proprietary languages for each database they wanted to connect to.

Through the ODBC Administrator in Control Panel, you can specify the particular database that is associated with a data source that an ODBC application program is written to use.

JDBC

In an effort to set an independent database standard API for Java, Sun Microsystems developed Java Database Connectivity, or JDBC. JDBC offers a generic SQL database access mechanism that provides a consistent interface to a variety of RDBMSs. This consistent interface is achieved through the use of "plug-in" database connectivity modules, or drivers. If a database vendor wishes to have JDBC support, he or she must provide the driver for each platform that the database and Java run on.

5. 3. MODULES

The work is divided into following modules:

- Network construction
- Synchronous delivery
- Asynchronous delivery
- Selecting best carrier

5.3.1. NETWORK CONSTRUCTION

This module is used to construct the partitioned network, and to enter the details of each user into the database. First we have to enter the number of hosts in each network. Then to register the host, for that the user must enter host name, port number, ip-address. After clicking the register button, it will check into the database whether that host details already exist or not. If the details are already present in the database it will display "host name must be unique" otherwise the details will be entered into the database. After that the partitioned network connections are to be established, it can be uni-direct or bi-direct. suppose we are choosing unidirection connection it will connect the first host to second host.and randomly allocate the weight for that connection.If we are choosing bidirection connection it will connect the first host to second host and also connect the second host to first host and also randomly allocate the weight for that connections. This should be done for each node in the network and for all networks.

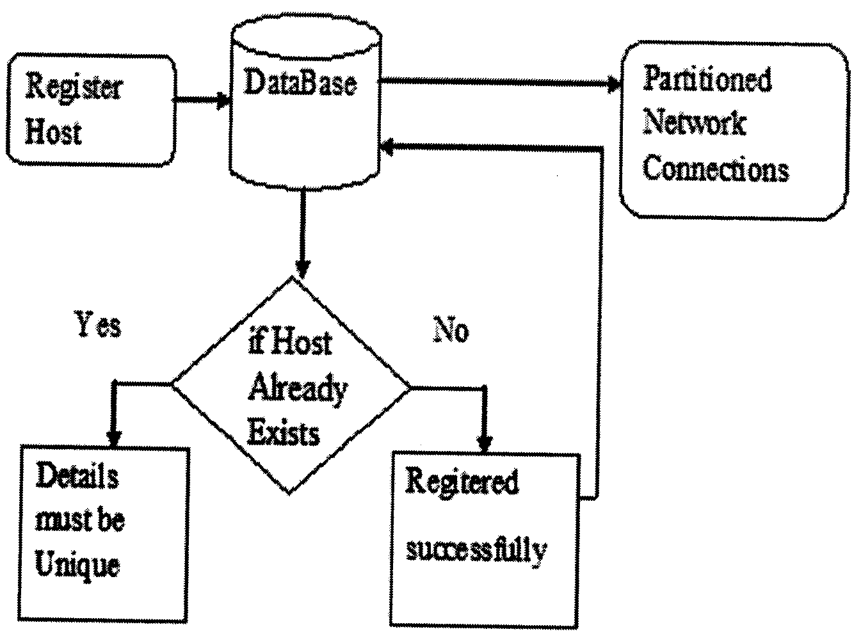


Fig.5.1.Network Construction module

5.3.2. SYNCHRONOUS DELIVERY

After constructing the partitioned network, login the hosts in two networks. Select the destination hostname from source hostname and select which message needs to be send from source to destination. After sending the message it checks synchronous delivery or asynchronous delivery based on network path. If the destination host is in the same network and a path exists between the source and destination its synchronous delivery, and its information will be present in the routing table of source. if message delivery is synchronous, it will find the shortest path to the destination using any of the routing protocols like Destination sequence distance vector routing(DSDV) or On-Demand multicast routing protocol(ODMRP). In that path source will directly send message to destination

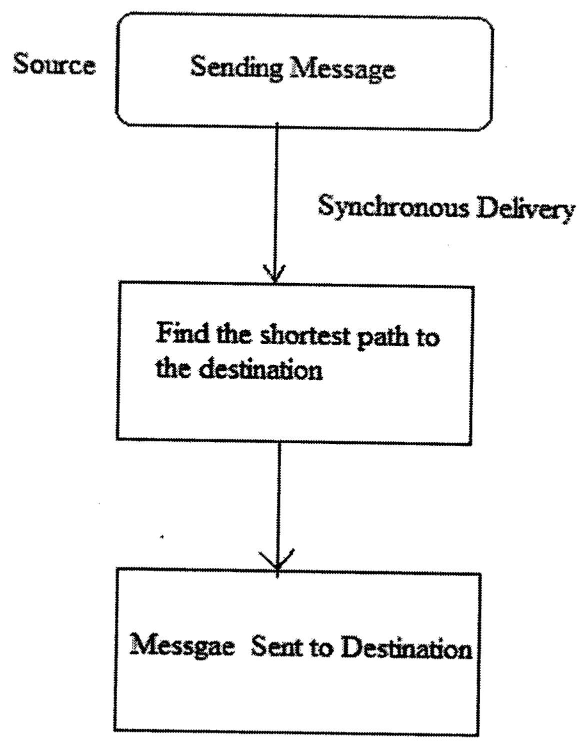


Fig 5.2.Synchronous delivery module

5.3.3. ASYNCHRONOUS DELIVERY

After constructing the partitioned network, login the hosts in two networks. Select the destination hostname from source hostname and select which message needs to be send from source to destination. Check whether the destination is in the same network (synchronous delivery) or into different network (asynchronous delivery).If the destination is in a different network as that of sender, calculate the delivery probability of each node to the destination node. The node which is having the highest delivery probability is the best carrier node. Send message to the carrier and the carrier will get detached from the network and join with the destination network as time goes. The message will be delivered to the destination node synchronously by finding the shortest path using any one of the routing protocol.

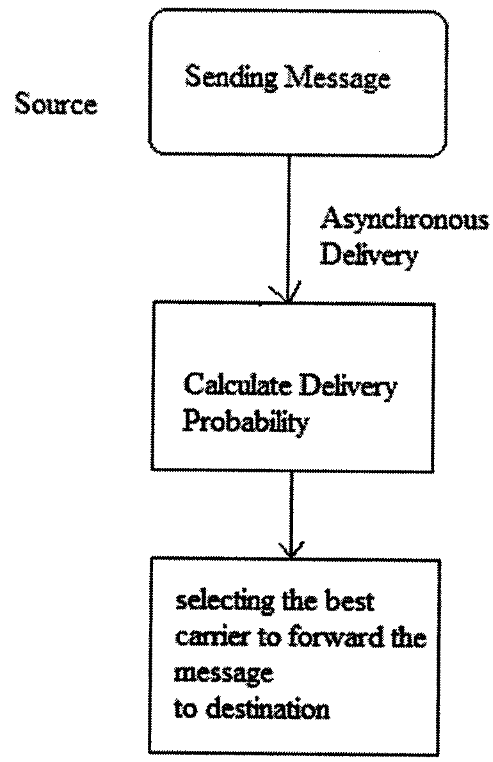


Fig.5.3.Asynchronous delivery module

5.3.4. SELECTING BEST CARRIER

In this module the best carrier host for asynchronous delivery of message is selected. Two attributes are considered for selecting the best carrier, mobility of the node and past collocation. For that calculate the delivery probability using utility function. Based on highest delivery probability select the best carrier host. After selecting best carrier host the message is stored in its buffer. After some time the node will disconnect from that network and connect to the destination network and transfer the message to the destination node.

In the implementation of Car we focus on two main attributes, the change degree of connectivity and the future host collocation, because these are the attributes most relevant to the ad hoc network.

The change degree of connectivity of host h is

$$Ucd_{ch}(t) = \frac{\text{Number of hosts that are neighbours in the time interval [t-T, t]}}{\text{Total number of hosts met in the time interval [t-T,t]}}$$

The collocation of host h with a host i is calculated as follows:

$$Ucol_{h,i}(t) = \begin{cases} 1 & \text{if the host h is collocated with i} \\ 0 & \text{Otherwise} \end{cases}$$

A value of 1 means that has been collocated with I at time t.

These values are fed into the Kalman filter predictors, which yield the predictions

$\hat{U}$ cdch and $\hat{U}$ col h,i of these utilities at time t+T.

These values are then composed into a single utility value using results from multicriteria decision theory described as follows,

$$U_{h,i} = w_{cdch}\,\hat{U}_{cdch} + w_{col\,h,i}\,\hat{U}_{col\,h,i}$$

Which represents how good of a node h is for delivering message to i.The weight *w* denote the relative importance of each attribute.The values of weights are the same for every host,ie the utility composition function is same for all nodes of the system.

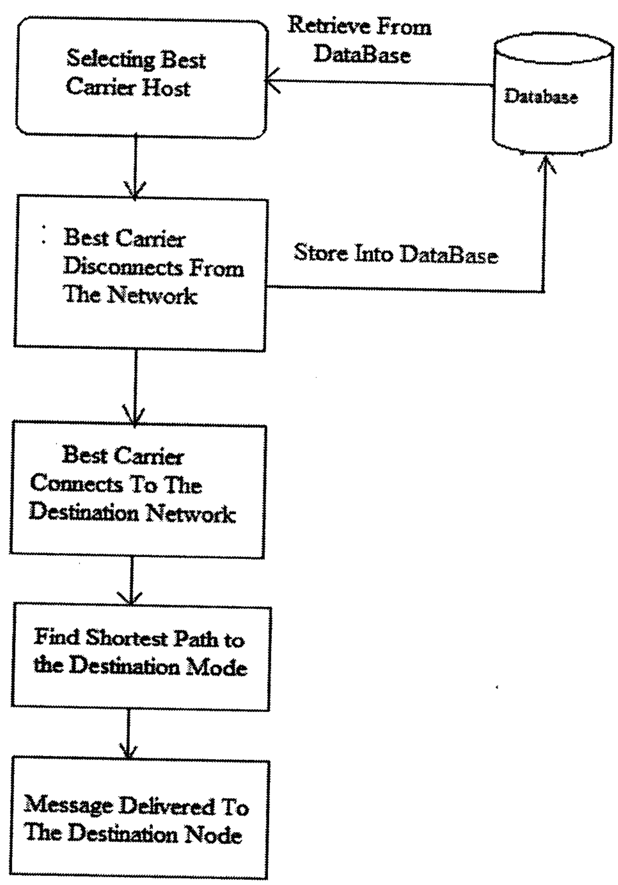


Fig 5.4.Best carrier selection module

The process of prediction and evaluation of the context information is as follows:

- Each host calculates its delivery probabilities for a given set of host. This is based on the calculation of utilities for each attribute describing the context. Then the future values of these utilities are predicted and composed using multicriteria decision theory to calculate overall delivery probability.
- The calculated delivery probabilities are periodically sent to the other hosts in the connected cloud as part of the update of routing information.
- Each host maintains a logical forwarding table of tuples describing the next logical hop and its associated delivery probability for all known destinations.
- Each host uses local prediction of delivery probabilities between updates of information. The prediction process is used during temporary disconnections and is carried out until certain accuracy can be guaranteed.

CHAPTER 6

EXPERIMENTAL RESULTS

The following graph shows the node usage while transferring the message from source to destination using mobility as a parameter

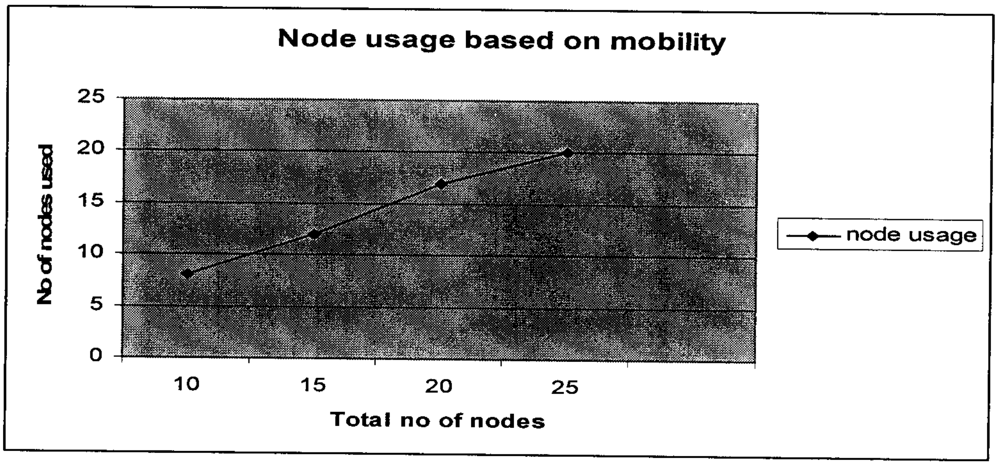


Fig.6.1.Node usage based on mobility

The following graph shows the node usage while transferring the message from source to destination using co-location as a parameter.

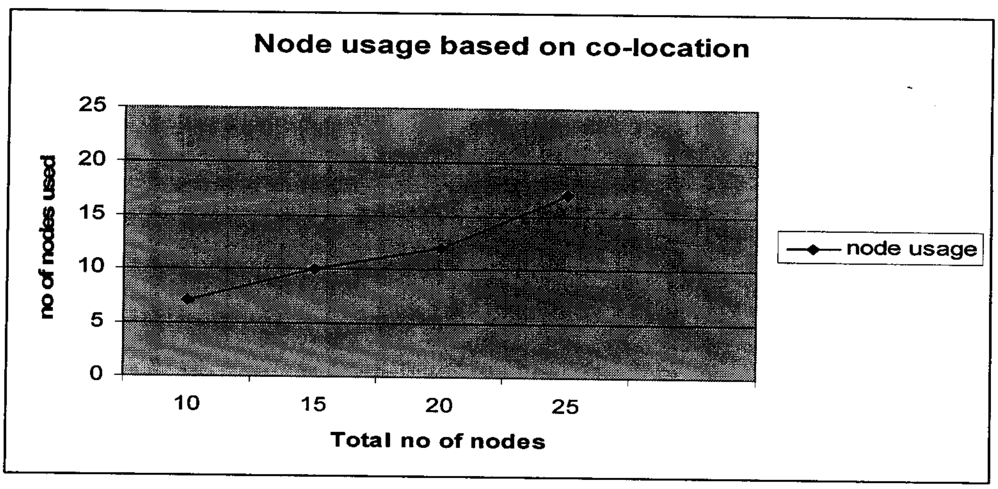


Fig.6.2.Node usage based on co-location

The following graph shows the node usage while transferring the message from source to destination using mobility and co-location as parameters.

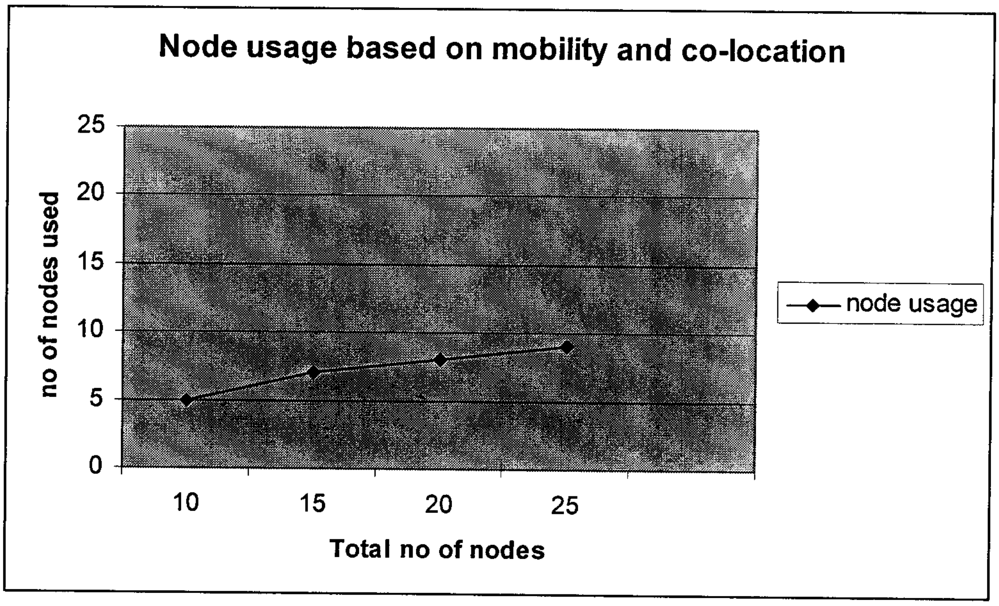


Fig.6.3.Node usage based on mobility and co-location

From the three graphs shown above Fig 6.1, Fig 6.2, Fig 6.3.It is clear that the last figure Fig 6.3 is using the least number of nodes compared to other two. From this we can infer that, if we are using mobility or co-location alone, the node usage for delivering the message to the destination will be higher. When we use the three parameters together, the number of nodes used for delivering the message to the destination will be least.

CHAPTER 7

Conclusion and Future Work

7.1. Conclusion

CAR provides communication in delay tolerant mobile ad hoc networks. Store-and-forward mechanisms are used to deliver messages in intermittently connected mobile ad hoc networks, where a connected path between the sender and receiver may not exist. Prediction technique and utility concepts are used to deliver message in intermittently connected mobile networks. While selecting carrier node battery power is considered. Data loss will never occur in CAR protocol as in previous technique because the battery power is also considered while selecting the carrier node.

7.2. Future Enhancement

In the feature work the selection of node can be based on more number of parameters like transmission range and computational ability. Further immediate plans include study of various mobility and failure environments. Further the buffer size can be considered infinity to avoid data loss.

APPENDIX I

SOURCE CODE

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.sql.*;
import java.net.*;
import java.io.*;
public class RegisterHost extends JFrame
{
        private JLabel lbl_host;
        private JLabel lbl_port;
        private JLabel lbl_ipadd;
        private JLabel lbl_title;
        private JTextField txt_host;
        private JTextField txt_port;
        private JTextField txt_ipadd;
        private JButton btn_reg;
        private JButton btn_cancel;
        private JPanel contentPane;
        String network="";
        String host;
        String ipadd,port,nodename1="",port1="",nodename2="",port2="";
        String str;
        Connection con;
        Statement stmt1;
        Statement stmt;
        ResultSet rs;
        Socket soc_reg;
        static int no;
        int count=0;
        public RegisterHost(int no)
        {

                super();
                try

               {
                        initializeComponent();
                        this.no=no;
                        this.setVisible(true);
```

```
this.setDefaultCloseOperation(EXIT_ON_CLOSE);
        }
        catch(Exception exp)
        {
exp.printStackTrace();
        }
}
private void initializeComponent()
{
        lbl_host = new JLabel();
        lbl_port = new JLabel();
        lbl_ipadd = new JLabel();
        lbl_title=new  JLabel("Context-Aware Adaptive Routing");
        txt_host = new JTextField();
        txt_port = new JTextField();
        txt_ipadd = new JTextField();
        btn_reg = new JButton();
        btn_cancel = new JButton();
        ButtonGroup bg=new ButtonGroup();
        contentPane = (JPanel)this.getContentPane();
        lbl_title.setFont(new Font("Times new roman",Font.BOLD,28));
        lbl_host.setFont(new Font("Monotype Corsiva",Font.BOLD,18));
        lbl_port.setFont(new Font("Monotype Corsiva",Font.BOLD,18));
        lbl_ipadd.setFont(new Font("Monotype Corsiva",Font.BOLD,18));
        btn_reg.setFont(new Font("Monotype Corsiva",Font.BOLD,18));
        btn_cancel.setFont(new Font("Monotype Corsiva",Font.BOLD,18));
        lbl_host.setText("HostName");
        lbl_port.setText("PortNo");
        lbl_ipadd.setText("IPAddress");
        txt_host.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e)
        {
                txt_host_actionPerformed(e);
        }

});
txt_port.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e)
{
        txt_port_actionPerformed(e);
}
```

```
        });
        txt_ipadd.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e)
        {
                txt_ipadd_actionPerformed(e);
        }
    });

    btn_reg.setText("Register");
    btn_reg.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e)
    {
            btn_reg_actionPerformed(e);
    }
    });
    btn_cancel.setText("Cancel");
    btn_cancel.addActionListener(new ActionListener()
    {
    public void actionPerformed(ActionEvent e)
    {
            btn_cancel_actionPerformed(e);
    }
    });
    contentPane.setLayout(null);
     addComponent(contentPane, lbl_host, 105,229,80,18);
    addComponent(contentPane, lbl_port, 104,291,80,18);
    addComponent(contentPane, lbl_ipadd, 106,365,80,18);
    addComponent(contentPane, txt_host, 201,229,119,22);
    addComponent(contentPane, txt_port, 201,291,119,22);
    addComponent(contentPane, txt_ipadd, 201,365,119,22);
    addComponent(contentPane, btn_reg, 100,427,95,28);
    addComponent(contentPane, btn_cancel, 271,427,95,28);
    addComponent(contentPane, lbl_title,25,177,500,20);
    this.setTitle("RegisterHost - extends JFrame");
    this.setLocation(new Point(0, 0));
    this.setSize(new Dimension(473, 539));
    }
    private void addComponent(Container container,Component c,int x,int y,int
    width,int height)

    {
                c.setBounds(x,y,width,height);
                container.add(c);
    }
    private void txt_host_actionPerformed(ActionEvent e)
    {
```

```
		System.out.println("\ntxt_host_actionPerformed(ActionEvent e)
		called.");
}
private void txt_port_actionPerformed(ActionEvent e)
{
		System.out.println("\ntxt_port_actionPerformed(ActionEvent e)
		called.");
}
private void txt_ipadd_actionPerformed(ActionEvent e)
{
		System.out.println("\ntxt_ipadd_actionPerformed(ActionEvent e)
		called.");
}
private void btn_reg_actionPerformed(ActionEvent e)
{
		String ipadd="",res="";
		System.out.println("\nbtn_reg_actionPerformed(ActionEvent e)
		called.");
		Try
		{
		host=txt_host.getText();
		port=txt_port.getText();
		ipadd=txt_ipadd.getText();
		sysname sn=new sysname();
		ipadd=sn.net();
		String ipadd1[]=ipadd.split("@");
		int hport=Integer.parseInt(ipadd1[1]);
		soc_reg=new Socket(ipadd1[0],hport);
		DataOutputStream dos_reg=new
		DataOutputStream(soc_reg.getOutputStream());
	      dos_reg.writeUTF("register");
	     dos_reg.writeUTF(host+"@"+port+"@"+ipadd);
		   DataInputStream dis_reg=new
	DataInputStream(soc_reg.getInputStream());
	  res=dis_reg.readUTF();
	  JOptionPane.showMessageDialog(null,res);
	  no--;
	  if(no==0)
	{

		this.setVisible(false);
		 new HostConnections();
	}

	}
	catch(Exception exp)
```

```
				{
					exp.printStackTrace();
				}
				finally
				{
					txt_host.setText("");
					txt_port.setText("");
					txt_ipadd.setText("");
				}
		  }
		private void btn_cancel_actionPerformed(ActionEvent e)
		{
		System.exit(0);

		}
		public static void main(String[] args)
		{
			new RegisterHost(no);
		}
}
```

LOGIN

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import com.jtattoo.plaf.mcwin.McWinLookAndFeel;
public class Login extends JFrame
{
	private JLabel lbl_top;
	private JLabel lbl_no;
	private JTextField txt_no;
	private JButton btn_Submit;
	private JPanel contentPane;
	public Login()
	{
		super();
		initializeComponent();
```

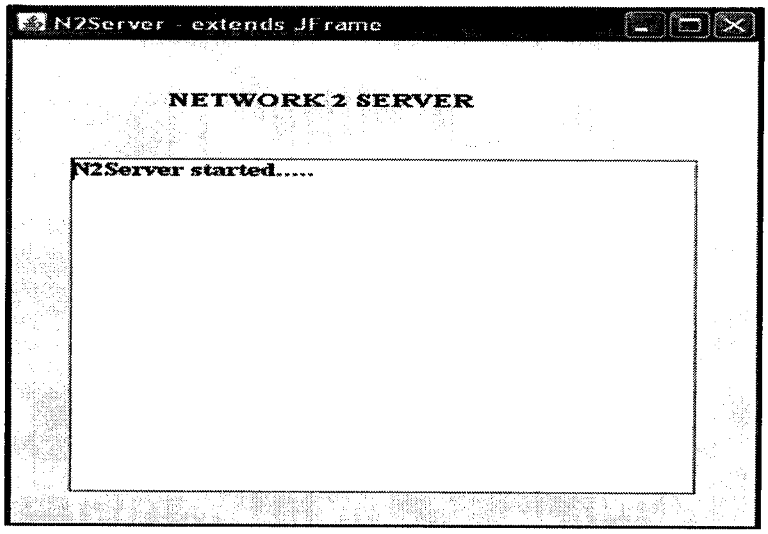
```
        this.setVisible(true);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
}


private void initializeComponent()
{
        lbl_top = new JLabel();
        lbl_no = new JLabel();
        txt_no = new JTextField();
        btn_Submit = new JButton();
        contentPane = (JPanel)this.getContentPane();
        lbl_top.setText("PARTITIONEDNETWORK  CONSTRUCTION");
        lbl_top.setFont(new Font("Times New Roman",Font.BOLD,13));
        lbl_top.setForeground(new Color(204, 102, 0));
        lbl_no.setText("Enter No.Of Hosts");
        lbl_no.setFont(new Font("Monotype Corsiva",Font.BOLD,15));
        lbl_no.setForeground(new Color(204, 102, 0));
        btn_Submit.setFont(new Font("Monotype Corsiva",Font.BOLD,18));
        btn_Submit.setForeground(new Color(204, 102, 0));
        txt_no.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e)
                {
                        txt_no_actionPerformed(e);
                }

        });
        btn_Submit.setText("Submit");
        btn_Submit.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e)
                {
```

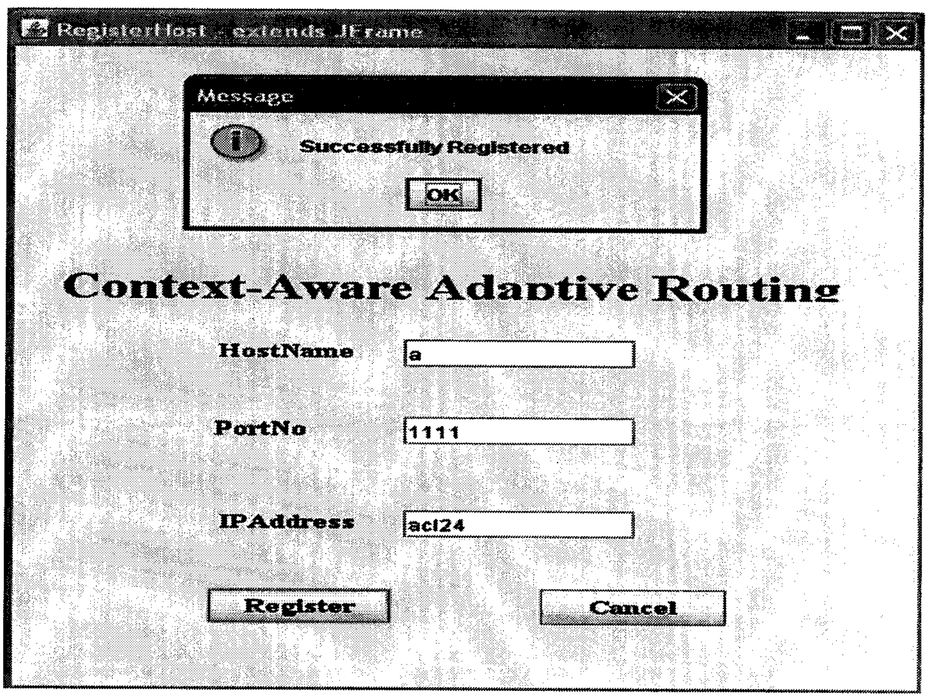
```
                        btn_Submit_actionPerformed(e);
                }

            });
            contentPane.setLayout(null);
            addComponent(contentPane, lbl_top, 80,57,270,18);
            addComponent(contentPane, lbl_no, 80,132,150,18);
            addComponent(contentPane, txt_no, 230,132,100,22);
            addComponent(contentPane, btn_Submit, 153,217,83,28);
            // Login
            this.setTitle("Login - extends JFrame");
            this.setLocation(new Point(0, 0));
            this.setSize(new Dimension(417, 346));
        }

        private void addComponent(Container container,Component c,int x,int
y,int width,int height)
        {
            c.setBounds(x,y,width,height);
            container.add(c);
        }

        private void txt_no_actionPerformed(ActionEvent e)
        {
            System.out.println("\ntxt_no_actionPerformed(ActionEvent e)
called.");
        }
        private void btn_Submit_actionPerformed(ActionEvent e)
        {
            System.out.println("\nbtn_Submit_actionPerformed(ActionEvent e)
called.");
```

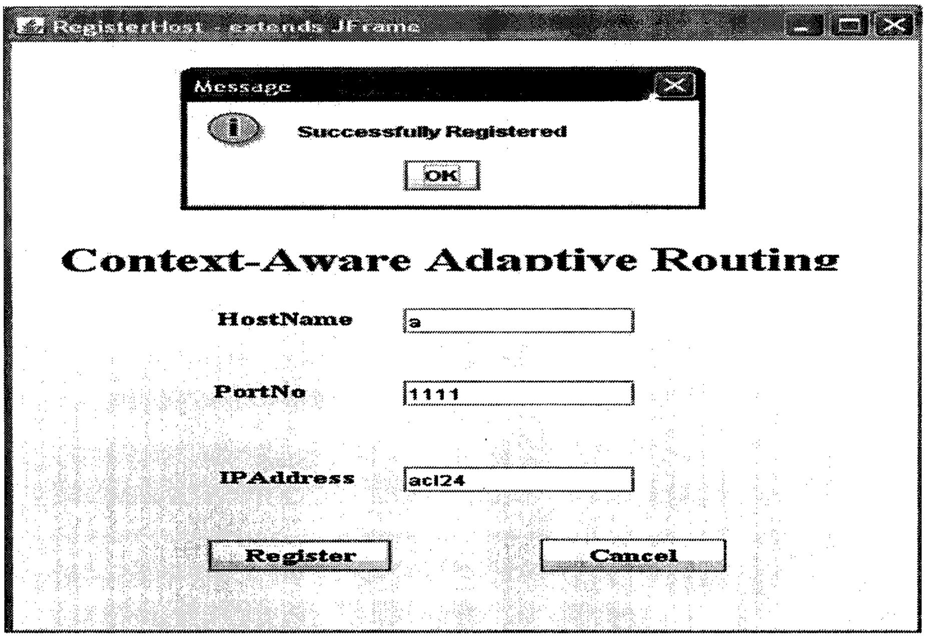
```
        this.setVisible(false);
        int no=Integer.parseInt(txt_no.getText());
        new RegisterHost(no);
    }
    public static void main(String[] args)
    {
        JFrame.setDefaultLookAndFeelDecorated(true);
        JDialog.setDefaultLookAndFeelDecorated(true);
        try
        {
UIManager.setLookAndFeel("com.jtattoo.plaf.mcwin.McWinLookAndFeel");
        }
        catch (Exception ex)
        {
            System.out.println("Failed loading L&F: ");
            System.out.println(ex);
        }
        new Login();
    }
```

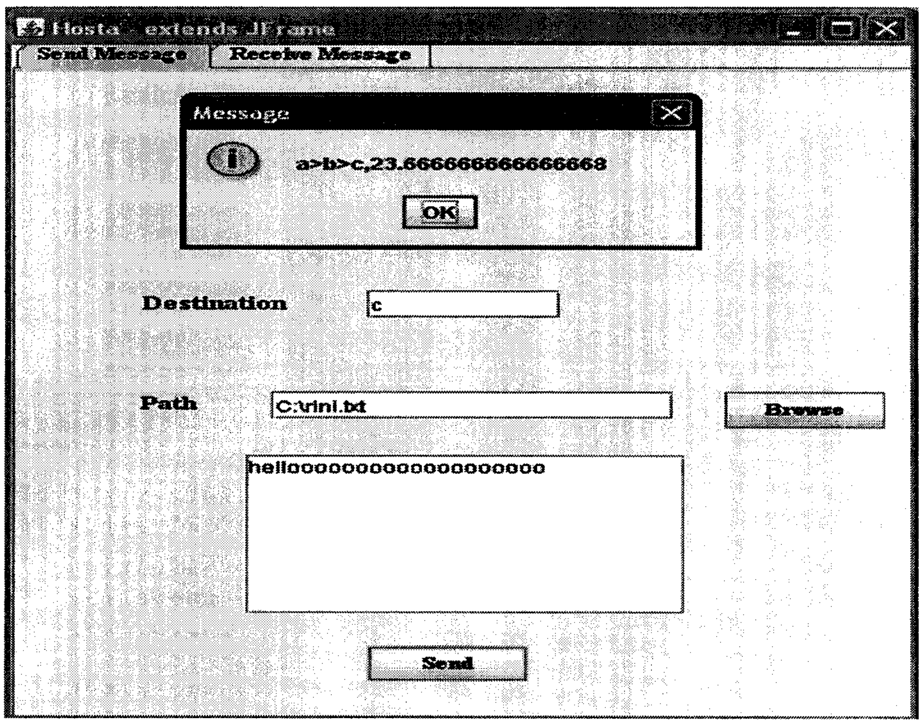
APPENDIX II

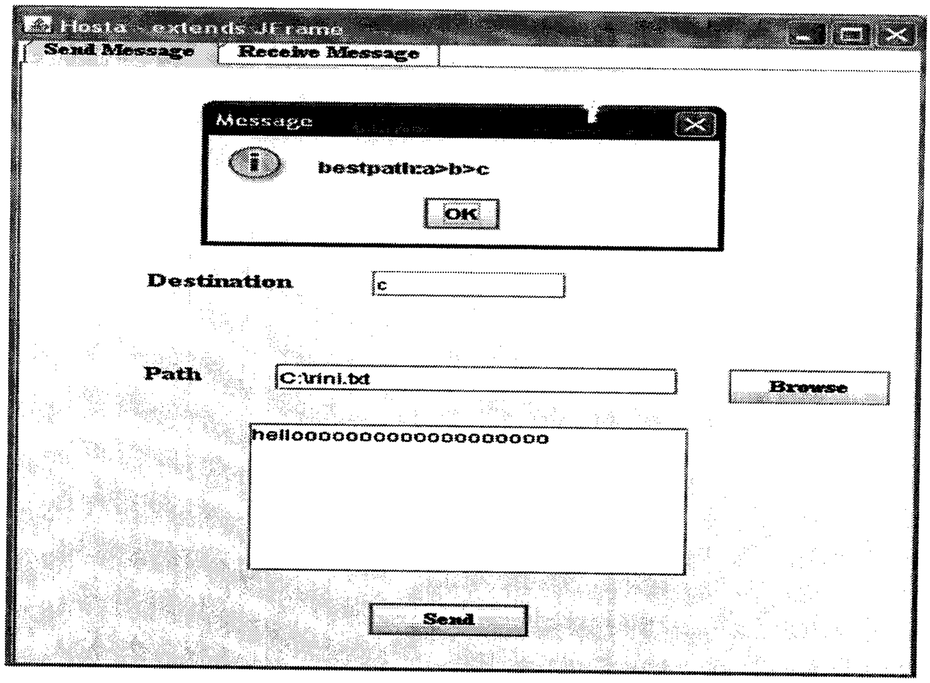
SNAPSHOTS

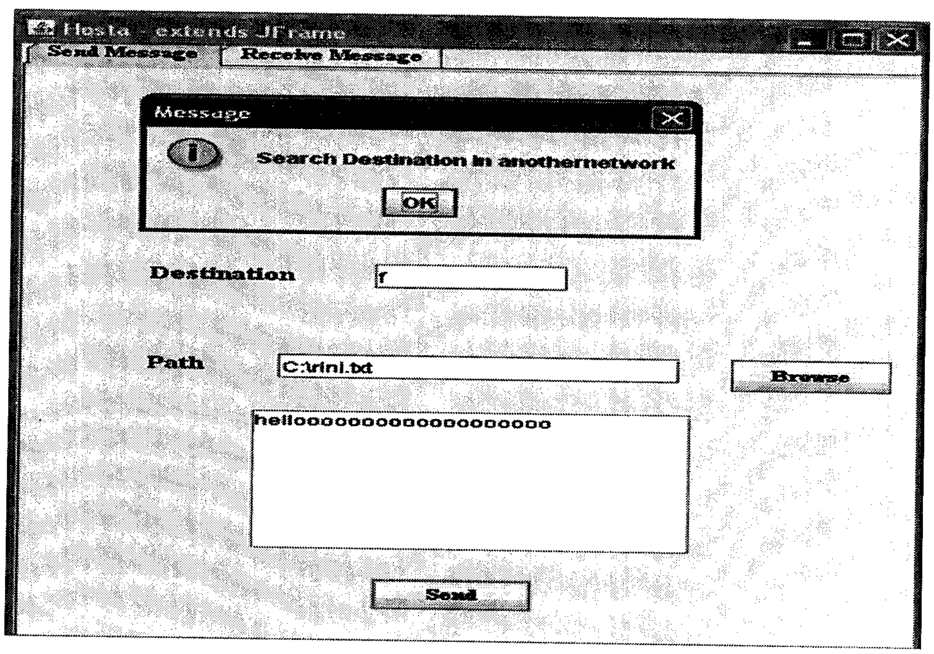


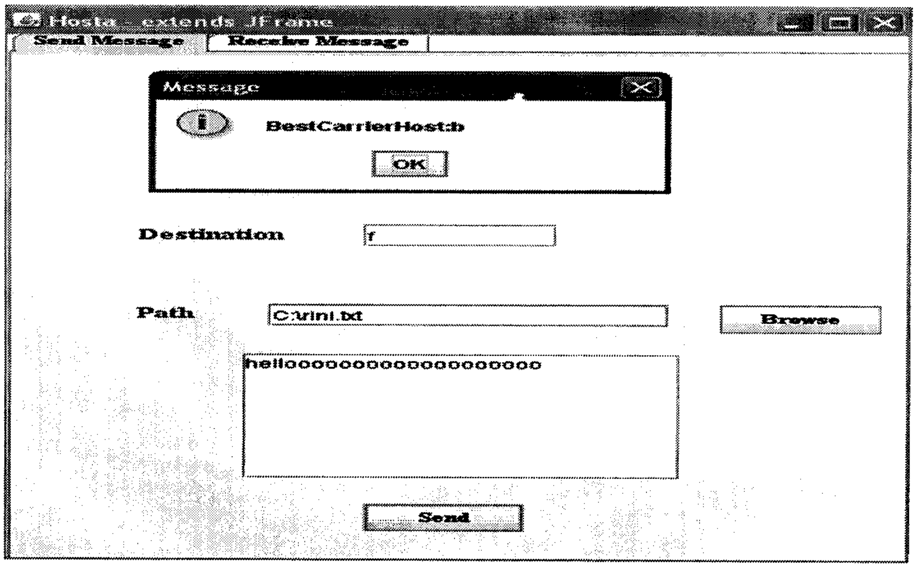


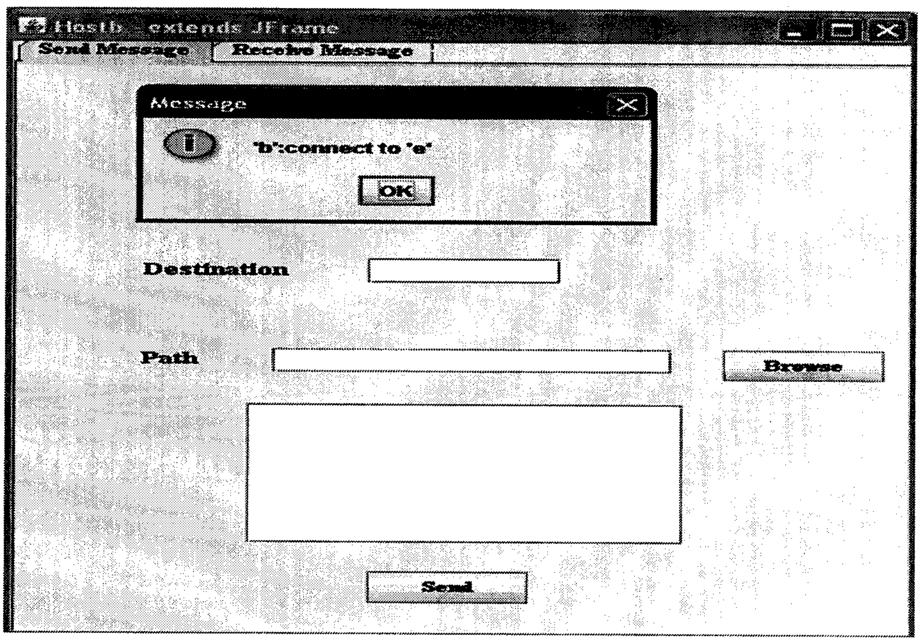

RegisterHost - extends JFrame
Message
Successfully Registered
OK
Context-Aware Adaptive Routing
HostName
a
PortNo
1111
IPAddress
acl24
Register
Cancel


Hosta - extends JFrame
Send Message
Receive Message
Message
a>b>c,23.666666666666668
OK
Destination
c
Path
C:\rini.txt
Browse
hellooooooooooooooooooo
Send


Hosta - extends JFrame
Send Message
Receive Message
Message
bestpath:a>b>c
OK
Destination
c
Path
C:\rini.txt
Browse
hellooooooooooooooooooooo
Send


Hosta - extends JFrame
Send Message
Receive Message
Message
Search Destination in anothernetwork
OK
Destination
r
Path
C:\rini.txt
Browse
hellooooooooooooooooooooo
Send


Hosta - extends JFrame
Send Message
Receive Message
Message
BestCarrierHost:b
OK
Destination
f
Path
C:\rini.txt
Browse
hellooooooooooooooooooo
Send


Hostb - extends JFrame
Send Message
Receive Message
Message
'b':connect to 'e'
OK
Destination
Path
Browse
Send

REFERENCES

[1] Mirco Musolesi,Cecilia "Mascolo,"CAR:Context-Aware Adaptive Routing For Delay-Tolerant Mobile Networks",IEEE TRANSACTIONS ON MOBILE COMPUTING,VOL. 8,NO.2,FEBRUARY 2009.

[2] A. Vahdat and D. Becker, "Epidemic Routing for Partially Connected Ad Hoc Networks", Technical Report CS-2000-06, Dept. Computer Science, Duke Univ., 2000.

[3] M.P.N. Sarafijanovic-Djukic and M. Grossglauser, "Island Hopping: Efficient Mobility Assisted Forwarding in Partitioned Networks", Proc. Third Ann. IEEE Conf. Sensor and Ad Hoc Comm. and Networks (SECON '06), Sept. 2006.

[4] Charles E Perkins,Pravin Bhagwat,"Highly Dynamic Destination-Sequenced Distance-Vector Routing(DSDV) for Mobile Computers", Proc. ACM SIGCOMM '07, pp. 300-315, Aug. 2005.

[5] M. Musolesi and C. Mascolo, "Evaluating Context Information Predictability for Autonomic Communication", Proc. IEEE Workshop Autonomic Comm. and Computing (ACC '06), June 2006.

[6] Z. Zhang, "Routing in Intermittently Connected Mobile Ad Hoc Networks and Delay Tolerant Networks: Overview and Challenges", IEEE Comm. Surveys and Tutorials, vol. 8, no. 1, Jan. 2006.

[7] J. Leguay, T. Friedman, and V. Conan, "Evaluating Mobility Pattern Space Routing for DTNs," Proc. IEEE INFOCOM '06, Apr. 2006.

[8] A. Balasubramanian, B.N. Levine, and A. Venkataramani, "DTN Routing as a Resource Allocation Problem," Proc. ACM SIGCOMM '07, pp. 373-384, Aug. 2007.

[9] S. Jain, K. Fall, and R. Patra, "Routing in a Delay Tolerant Network," Proc. ACM SIGCOMM '04, Aug. 2004.

[10] V. Zhao, M. Ammar, and E. Zegura, "A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks," Proc. ACM MobiHoc '04, May 2004.

[11] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic Routing inIntermittently Connected Networks," Mobile Computing and Comm. Rev., vol. 7, no. 3, July 2003.

[12] R.E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," Trans. ASME J. Basic Eng., Mar. 1960.

[13] Sung-Ju Lee, Mario Gerla, and Ching-Chuan Chiang," On-Demand Multicast Routing Protocol", Defense Advanced Research Projects Agency (DARPA),May 2000